



⑪ BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ **Offenlegungsschrift**  
⑩ **DE 197 04 728 A 1**

⑤ Int. Cl.<sup>6</sup>:  
**G 06 F 15/80**  
G 06 F 1/04

⑲ Aktenzeichen: 197 04 728.9  
⑳ Anmeldetag: 8. 2. 97  
㉑ Offenlegungstag: 13. 8. 98

DE 197 04 728 A 1

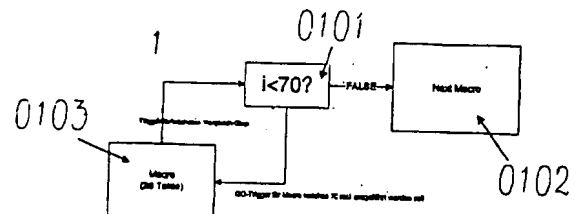
⑦ Anmelder:  
Pact Informationstechnologie GmbH, 81545  
München, DE

⑧ Erfinder:  
Vorbach, Martin, 76149 Karlsruhe, DE; Münch,  
Robert, 76149 Karlsruhe, DE

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

⑤ Verfahren zur Selbstsynchronisation von konfigurierbaren Elementen eines programmierbaren Bausteines

⑤ In Verbindung mit einem Verfahren zur Synchronisation und Umkonfiguration von konfigurierbaren Elementen in Bausteinen mit zwei- oder mehrdimensionaler programmierbarer Zellstruktur (DFP, FPGA, DPGA) wird vorgeschlagen, daß die Synchronisationssignale während der Verarbeitung innerhalb des Datenstromes von den verarbeitenden konfigurierbaren Elementen generiert und an weitere Elemente zur Synchronisation über den Datenbus gesandt werden und daß aus dem Datenstrom heraus anhand entsprechender Befehle Konfigurationswörter innerhalb eines programmierbaren Elementes generiert und über den Datenbus zusammen mit der Adresse des anzusprechenden Registers an ein weiteres konfigurierbares Element übertragen werden, wobei dieses dadurch ohne den Einfluß einer externen Ladelogik (um)konfiguriert wird.



DE 197 04 728 A 1

## Beschreibung

## 1 Hintergrund der Erfindung

## 1.1 Stand der Technik

## 1.2 Probleme

Bei heutigen Bausteinen (FPGA, DPGA etc.) wird die Synchronisation der konfigurierbaren Elemente meistens durch den Takt des Bausteines hergestellt. Diese Art der zeitlich gesteuerten Synchronisation bereitet viele Probleme, da oft nicht im Vorhinein bekannt ist, wie lange eine Aufgabe benötigt, bis ein gültiges Ergebnis bereit steht. Ein weiteres Problem der zeitgesteuerten Synchronisation ist, daß das Ereignis auf welches die Synchronisation erfolgt nicht von dem zu synchronisierenden Element selbst ausgelöst wird, sondern von einem unabhängigen Element. In diesem Fall sind nun zwei verschiedene Elemente an der Synchronisation beteiligt. Dies führt zu einem erheblich höherem Verwaltungsaufwand.

## 1.3 Verbesserung durch die Erfindung

Durch die Erfindung wird ein Verfahren beschrieben, welches es gestattet, daß die Synchronisation von zu synchronisierenden Elementen selbst ausgeht. Die Synchronisation ist nicht mehr durch eine zentrale Instanz implementiert und wird auch nicht mehr durch eine zentrale Instanz verwaltet. Durch die Verlegung der Synchronisation in jedes Element können auch viel mehr Synchronisationsaufgaben gleichzeitig durchgeführt werden, da unabhängige Elemente sich nicht mehr gegenseitig beim Zugriff auf die zentrale Synchronisations-Instanz behindern. Die Einzelheiten und besondere Ausgestaltungen, sowie Merkmale des erfindungsgemäßen Synchronisationsverfahrens sind Gegenstand der Patentansprüche.

## 2 Beschreibung der Erfindung

## 2.1 Übersicht über die Erfindung, Abstrakt

In einem Baustein mit zwei- oder mehrdimensional angeordneter, programmierbarer Zellstruktur (DFP, DPGA) kann jedes konfigurierbare Element über eine Vernetzungsstruktur auf die Konfigurations- und Statusregister der anderen konfigurierbaren Elemente zugreifen und damit deren Funktion und Arbeitsweise aktiv beeinflussen. Die Konfiguration kann somit zusätzlich zu der üblichen Methode durch eine Ladelogik aus dem ProcessingArray (PA vgl. PACT02) heraus erfolgen.

## 2.2 Detailbeschreibung der Erfindung

Es wird von einem frei zur Laufzeit programmierbaren Baustein ausgegangen, welcher zusätzlich zur Laufzeit rekonfiguriert werden kann. Die auf dem Chip enthaltenen konfigurierbaren Elemente besitzen ein oder mehrere Konfigurationsregister für verschiedene Aufgaben. Auf diese Konfigurationsregister kann lesend wie schreibend zugegriffen werden. In dem beschriebenen Verfahren wird davon ausgegangen, daß für folgende Informationen eine Konfiguration in einem zu konfigurierenden Element eingestellt werden kann.

- Vernetzungs-Register. In diesem Register wird die Art der Verbindung zu anderen Zellen eingestellt.
- Befehls-Register. In diesem Register wird die auszu-

führende Funktion des konfigurierbaren Elements eingetragen.

- Status-Register. In diesem Register speichert die Zelle ihren aktuellen Zustand. Dieser Zustand gibt anderen Elementen des Bausteins Auskunft darüber, in welchem Verarbeitungszyklus sich die Zelle befindet.

Eine Zelle wird durch einen Befehl konfiguriert, welcher die Funktion der Zelle bestimmt, die ausgeführt werden soll. Weiterhin werden Konfigurationsdaten eingetragen um die Vernetzung mit anderen Zellen und den Inhalt des Status-Registers einzustellen. Nach diesem Vorgang ist die Zelle betriebsbereit.

Um eine flexible und dynamische Zusammenarbeit vieler Zellen zu ermöglichen, kann jede Zelle auf alle Konfigurationsregister einer anderen Zelle lesend oder schreibend zugreifen. Auf welches der vielen Konfigurationsregister lesend oder schreibend zugegriffen wird, wird durch die Art des Befehls, mit welchem die Zelle konfiguriert wurde, festgelegt. Jeder Befehl den die Zelle ausführen kann, existiert in soviel verschiedenen Adressierungsarten, wie es verschiedene, voneinander unabhängige Konfigurationsregister, in einem zu konfigurierenden Element gibt.

Beispiel: Eine Zelle besitzt die oben angegebenen Konfigurationsregister (Vernetzung, Befehl und Status) und soll den Befehl ADD, welcher eine Addition durchführt ausführen. Durch die verschiedenen Arten des ADD Befehls kann nun selektiert werden, wohin das Ergebnis dieser Funktion übertragen wird.

- ADD-A. Das Ergebnis wird an das Operand-Register-A der Zielzelle übertragen.
- ADD-B. Das Ergebnis wird an das Operand-Register-B der Zielzelle übertragen.
- ADD-V. Das Ergebnis wird an das Vernetzungs-Register der Zielzelle übertragen.
- ADD-S. Das Ergebnis wird an das Status-Register der Zielzelle übertragen.
- ADD-C. Das Ergebnis wird an das Befehls-Register der Zielzelle übertragen.

Neben dem Ergebnis kann jede Zelle eine Menge an Trigger-Signalen erzeugen. Die Trigger-Signale müssen nicht notwendigerweise an die gleiche Zielzelle übertragen werden, wie das Ergebnis der Verarbeitung des konfigurierten Befehls. Ein Trigger-Signal oder erst die Kombination mehrerer Trigger-Signale, löst bei der Zielzelle eine bestimmte Aktion aus oder setzt die Zelle in einen bestimmten Zustand. Eine Beschreibung der Zustände ist weiter unten im Text zu finden. Folgende Trigger-Signale gibt es:

- GO-Trigger. Der GO-Trigger setzt die Zielzelle in den Zustand READY.
- RECONFIG-Trigger. Der RECONFIG-Trigger setzt die Zielzelle in den Zustand RECONFIG, so daß die Zelle umprogrammiert werden kann. Besonders in Zusammenarbeit mit Switching-Tabellen ist dieser Trigger sehr sinnvoll. Geht man davon aus, daß zu verarbeitenden Daten mit der steigenden Taktflanke in die Operanden-Register geladen werden, in der Zeitspanne des H-Level verarbeitet werden und mit der fallenden Flanke in das Ausgangsregister geschrieben werden, so ist eine Rekonfiguration der Zelle mit der fallenden Flanke möglich. Mit der fallenden Flanke werden die neuen Konfigurationsdaten in das Befehls-Register geschrieben. Die Zeitspanne des L-Level ist ausreichend genug, um die Rekonfiguration erfolgreich abzuschließen.

- STEP-Trigger. Der STEP-Trigger löst bei der Zielzelle, welche sich im Zustand WAIT befindet, die einmalige Ausführung des konfigurierten Befehls aus.
- STOP-Trigger. Der STOP-Trigger hält die Zielzelle an, in dem die Zelle in den Zustand STOP gesetzt wird.

Durch die Möglichkeit in der verarbeitenden Zelle anzugeben, in welches Register der Zielzelle das Ergebnis eingetragen werden soll und welche Art von Trigger-Signal erzeugt werden soll, kann aus einem Datenstrom eine Menge an Verwaltungsdaten erzeugt werden. Diese Verwaltungsdaten stellen kein Ergebnis der eigentlichen Aufgabe dar, welche durch den Chip abgearbeitet werden soll, sondern dienen allein der Verwaltung, Synchronisation, Optimierung etc. des internen Zustands.

Jede Zelle kann folgende Zustände annehmen, welche durch eine geeignete Kodierung im Status-Register dargestellt werden.

- READY. Die Zelle ist mit einem gültigen Befehl konfiguriert worden und kann Daten verarbeiten. Die Verarbeitung findet mit jedem Taktzyklus statt. Die Daten werden auf Grund der Adressierungsart der datenschickenden Zelle in die Register der Zielzelle eingelesen.
- WAIT. Die Zelle ist mit einem gültigen Befehl konfiguriert worden und kann Daten verarbeiten. Die Verarbeitung findet mit auf Grund eines Trigger-Signals statt, welches durch andere Elemente des Bausteins erzeugt werden können. Die Daten werden auf Grund der Adressierungsart der datenschickenden Zelle in die Register der Zielzelle eingelesen.
- CONFIG. Die Zelle ist nicht mit einem gültigen Befehl konfiguriert. Das Datenpaket, welches mit dem nächsten Taktzyklus an die Zelle gesandt wird, wird in das Befehls-Register eingelesen. Das Datenpaket wird auf jeden Fall in das Befehls-Register eingelesen, egal welche Adressierungsart von der datenschickenden Zelle benutzt wurde.
- CONFIG-WAIT. Die Zelle ist nicht mit einem gültigen Befehl konfiguriert. Ein Datenpaket, wird mit dem nächsten Trigger-Signal, welches durch andere Elemente des Bausteins erzeugt werden kann, eingelesen und in das Befehls-Register geschrieben. Das Datenpaket wird auf jeden Fall in das Befehls-Register eingelesen, egal welche Adressierungsart von der datenschickenden Zelle benutzt wurde.
- RECONFIG. Die Zelle ist mit einem gültigen Befehl konfiguriert, verarbeitet aber keine weiteren Daten, nimmt die Daten auch nicht an. Die Zelle kann durch ein anderes Element des Bausteins umkonfiguriert werden.
- STOP. Die Zelle ist mit einem gültigen Befehl konfiguriert, verarbeitet aber momentan keine Daten. Die Daten werden von der Zelle angenommen (in die Eingangsregister übertragen), aber nicht weiterverarbeitet.

Durch diese verschiedenen Zustände und der Möglichkeit auf die verschiedenen Register einer Zelle schreibend und lesend zuzugreifen, kann jede Zelle eine aktive Verwaltungsrolle einnehmen. Im Gegensatz dazu besitzen alle existierenden Bausteine dieser Art eine zentrale Verwaltungsinstantz, welche immer den gesamten Zustand des Bausteins kennen und handhaben muß.

Um eine weitere Flexibilität zu erreichen gibt es eine weitere Klasse an Befehlen, die nach der ersten Ausführung ihre Art wechseln. Bezogen auf das Beispiel des ADD-Befehls sieht ein Befehl dann so aus:

- ADD-C-A. Das Ergebnis der ADD Funktion wird bei der ersten Ausführung des Befehls in das Befehls-Register der Zielzelle geschrieben. Bei jeder weiteren Ausführung wird das Ergebnis in das Operand-Register-A geschrieben.

Diese Möglichkeit kann beliebig erweitert werden, so daß auch Befehle der Art ADD(-) V-A(-) ...-B denkbar sind. Jeder Befehl kann alle permutierten Kombinationen der verschiedenen Adressierungs- und Trigger-Arten annehmen.

## 2.3 Erweiterung der Hardware gegenüber PACT02

### 2.3.1 Zusätzliche Register

Zu den in PACT02 beschriebenen Register kommt ein Statusregister und ein Konfigurationsregister hinzu. Beide Register werden vom PLU-Bus angesteuert und haben Verbindung zur Zustandsmaschine der SM-UNIT (PACT02 Fig. 2 0213).

### 2.3.2 Veränderung des PLU-Busses

In PACT02 werden die konfigurierbaren Register M-/PLUREG ausschließlich über den PLU-Bus (PACT02 Fig. 2 0210) verwaltet. Um die erfindungsgemäße Funktion zu gewährleisten muß nunmehr eine zusätzliche Zugriffsmöglichkeit durch den normalen Systembus (PACT02 Fig. 2 0201) möglich sein. Dasselbe gilt für die neuen Status- und Konfigurationsregister.

Dabei ist nur der Teil des Systembusses für die Register relevant, der über die BM-UNIT (PACT02 Fig. 2 0210) mit der PAE vernetzt ist. Daher wird der Bus von der BM-UNIT an die Register weitergeleitet, wo vorgeschaltete Multiplexer oder vorgeschaltete Tore die Umschaltung zwischen dem PLU-Bus und dem für die PAE relevanten Systembus übernehmen.

Dabei sind die Multiplexer oder Tore so geschaltet, daß sie immer den für die PAE relevanten Systembus durchschalten, außer nach einem Rücksetzen des Bausteins (RESET) oder wenn das ReConfig-Signal (PACT02 Fig. 3 0306) aktiv ist.

### 2.3.3 Erweiterungen des Systembusses

Der Systembus (PACT02 Fig. 2 0201) wird dahingehend erweitert, daß zusammen mit den Daten die Informationen über die Zielregister übertragen werden. Das bedeutet, eine Adresse wird mitgeschickt, die beim Datenempfänger das gewünschte Register selektiert.

## 3 Kurzbeschreibung der Diagramme

Fig. 1 Diese Figur zeigt, wie durch den Einsatz von Triggern ein Schleifenkonstrukt implementiert werden kann.

Fig. 2 Diese Figur zeigt, wie durch den Einsatz mehrerer Trigger ein Vergleichskonstrukt implementiert werden kann.

Fig. 3 Diese Figur zeigt, wie durch den Einsatz mehrerer Trigger und deren Verschachtelung ein Vergleichskonstrukt mit mehreren Ausgängen implementiert werden kann.

Fig. 4 zeigt die notwendigen Erweiterungen gegenüber PACT02.

## 4 Detailbeschreibung der Diagramme und Ausführungsbeispiele

Fig. 1 Das Makro 0103 soll in diesem Beispiel 70 mal ausgeführt werden. Eine Ausführung des Makros benötigt

26 Taktzyklen. Das bedeutet, daß nur alle 26 Taktzyklen der Zähler 0101 um eins verringert werden darf. Ein Problem bei frei programmierbaren Bausteinen ist nun, daß nicht immer garantiert werden kann, daß auch wirklich nach 26 Taktzyklen die Abarbeitung des Makros 0103 abgeschlossen ist. Eine Verzögerung kann zum Beispiel dadurch entstehen, daß ein Makro, welches die Eingangsdaten für Makro 0103 liefern soll, plötzlich 10 Taktzyklen länger benötigt. Aus diesem Grund sendet die Zelle in Makro 0103 ein Trigger-Signal an den Zähler 0101, durch welche das Ergebnis der Berechnung an ein weiteres Makro gesandt wird. Gleichzeitig wird die Verarbeitung des Makros 0103 durch die gleiche Zelle gestoppt. Diese Zelle 'weis' genau, daß die Bedingung für die Beendigung einer Berechnung erreicht wurde.

Das gesendete Trigger-Signal ist in diesem Fall ein STEP-Trigger, welcher veranlaßt, daß der Zähler 0101 einmal seine konfigurierte Funktion ausführt. Der Zähler zählt seinen Zählerwert um eine herunter und vergleicht, ob er den Wert 0 erreicht hat. Ist dies nicht der Fall, wird ein GO-Trigger an das Makro 0103 abgeschickt. Dieses GO-Trigger-Signal veranlaßt das Makro 0103 seine Funktion wieder aufzunehmen.

Dieser Vorgang wiederholt sich solange, bis der Zähler 0101 den Wert 0 erreicht hat. In diesem Fall wird ein Trigger-Signal an das Makro 0102 geschickt und löst dort eine Funktion aus.

Durch dieses Zusammenspiel von Triggern kann eine sehr feingranulare Synchronisation erreicht werden.

Fig. 2 Fig. 2 entspricht der Grundidee her Fig. 1. Die Funktion in Element 0202 ist diesmal jedoch kein Zähler sondern ein Vergleichser. Das Makro 0201 schickt nach jedem Verarbeitungsdurchlauf einen Vergleichswert mit an den Vergleichser 0202. Je nach Ausgang des Vergleichs, werden wiederum verschiedene Trigger angesteuert um zum Beispiel eine Aktion in den Makros 0203 zu veranlassen. Das in Fig. 2 implementierte Konstrukt entspricht dem einer IF-Abfrage in einer Programmiersprache.

Fig. 3 Wie in Fig. 2 werden hier mehrere Vergleichser 0301, 0302 eingesetzt, um die Konstruktion eines IF-ELSE-ELSE Konstruktes (oder einer Mehrfachauswahl) zu implementieren. Durch die Verwendung verschiedenster Arten von Triggern und Verbindungen dieser Trigger zu den Makros 0303, 0304 können sehr komplexe Abläufe einfach implementiert werden.

Fig. 4 zeigt die Unterschiede zu PACT02 Fig. 2. Dabei sind die Unterschiede schattiert eingetragen. Das Konfigurationsregister (0401) und das Statusregister (0402) haben über den Bus (0407) Verbindung zur SM-UNIT. Die Register 0401, 0402, F- und M-PLUREG sind über einen internen Bus 0206 mit einem Tor 0403 verbunden. Dieses verbindet den internen Bus (0406) je nach Stellung mit dem PLU-Bus 0405 um eine Konfiguration durch die PLU zu ermöglichen oder über einen Bus 0408 mit dem BM-UNIT. Diese schaltet je nach Adressierung auf dem Datenbus 0404 die Daten zu den O-REG weiter oder zu dem adressierten Register 0401, 0402, F- oder M-PLUREG.

#### 6 Begriffsdefinition

**BM-UNIT** Einheit zum Aufschalten der Daten auf die Bussysteme außerhalb der PAE. Das Aufschalten geschieht über Multiplexer für die Dateneingänge und Tore für die Datenausgänge. oACK-Leitungen sind als Open-Kollektor-Treiber implementiert. Die BM-UNIT wird durch das M-PLUREG gesteuert.

**Dateneempfänger** Die Einheit(en), die Ergebnisse der PAE weiterverarbeitet/arbeiten

**Datensender** Die Einheit(en), die Daten für die PAE als Ope-

randen zur Verfügung stellt/stellen

**Datenwort** Ein Datenwort besteht aus einer beliebig langen Bit-Reihe. Diese Bit-Reihe stellt eine Verarbeitungseinheit für eine Anlage dar. In einem Datenwort können sowohl Befehle für Prozessoren o. ä. Bausteine sowie rein Daten kodiert werden.

**DFP** Datenflußprozessor nach Patent/Offenlegung DE 44 16 881

**DPGA** Dynamisch konfigurierbare FPGAs. Stand der Technik

**EALU** Erweiterte arithmetisch logische Einheit. ALU, die um Sonderfunktionen, die zum Betrieb einer Datenverarbeitungseinrichtung gemäß DE 44 16 881 A1 benötigt werden oder sinnvoll sind erweitert wurde. Dies sind insbesondere

**Zähler.** Elemente Sammelbegriff für alle Arten von in sich abgeschlossenen Einheiten, welche als Stück in einem elektronischen Baustein zum Einsatz kommen können. Elemente sind also:

- Konfigurierbare Zellen aller Art
- Cluster
- RAM-Blöcke
- Logik
- Rechenwerke
- Register
- Multiplexer
- I/O Pins eines Chips

**Ereignis** Ein Ereignis kann durch ein Hardwareelement in irgendeiner zur Anwendung passenden Art und Weise ausgewertet werden und als Reaktion auf diese Auswertung eine bedingte Aktion auslösen. Ereignisse sind somit zum Beispiel:

- Taktzyklus einer Rechanlage.
- internes oder externes Interrupt-Signal.
- Trigger-Signal von anderen Elementen innerhalb des Bausteines.
- Vergleich eines Datenstroms und/oder eines Befehlsstroms mit einem Wert.
- Input/Output Ereignisse.
- Ablaufen, überlaufen, neusetzen etc. eines Zählers.
- Auswerten eines Vergleichs.

**FPGA** Programmierbarer Logikbaustein. Stand der Technik. F-PLUREG Register in dem die Funktion der PAE gesetzt wird. Ebenfalls wird der OneShot- und Sleep-Mode gesetzt. Das Register wird von der PLU beschrieben.

**H-Pegel** Logisch 1 Pegel, abhängig von der verwendeten Technologie

**konfigurierbares Element** Ein konfigurierbares Element stellt eine Einheit eines Logik-Bausteines dar, welche durch ein Konfigurationswort für eine spezielle Funktion eingestellt werden kann. Konfigurierbare Elemente sind somit, alle Arten von RAM-Zellen, Multiplexer, Arithmetische logische Einheiten, Register und alle Arten von interner und externer Vernetzungsbeschreibung etc.

**konfigurierbare Zelle** Siehe Logikzellen

**Konfigurieren** Einstellen der Funktion und Vernetzung einer logischen Einheit, einer (FPGA)-Zelle oder einer PAE (vgl. unkonfigurieren).

**Konfigurationsdaten** Beliebige Menge von Konfigurationsworten.

**Konfigurationsspeicher** Der Konfigurationsspeicher enthält ein oder mehrere Konfigurationsworte.

**Konfigurationswort** Ein Konfigurationswort besteht aus einer beliebig langen Bit-Reihe. Diese Bit-Reihe stellt eine

gültige Einstellung für das zu konfigurierende Element dar, so daß eine funktionsfähige Einheit entsteht.

Ladelogik Einheit zum Konfigurieren und Umkonfigurieren der PAE. Ausgestaltet durch einen speziell an seine Aufgabe angepaßten Mikrokontroller.

Logikzellen Bei DFPs, FPGAs, DPGAs verwendete konfigurierbare Zellen, die einfache logische oder arithmetische Aufgaben gemäß ihrer Konfiguration erfüllen.

L-Pegel Logisch 0 Pegel, abhängig von der verwendeten Technologie

M-PLUREG Register in dem die Vernetzung der PAE gesetzt wird. Das Register wird von der PLU beschrieben.

O-REG Operandenregister zur Speicherung der Operanden der EALU. Ermöglicht die zeitliche und funktionelle Unabhängigkeit der PAE von den Datensendern. Dadurch wird der Transfer der Daten vereinfacht, da er asynchron oder pakteriorientiert stattfinden kann. Gleichzeitig wird die Möglichkeit geschaffen die Datensender unabhängig von der PAE oder die PAE unabhängig von den Datensendern umzu-konfigurieren.

PLU Einheit zum Konfigurieren und Umkonfigurieren der PAE. Ausgestaltet durch einen speziell an seine Aufgabe angepaßten Mikrokontroller.

SM-UNIT StateMachine-UNIT. Zustandsmaschine, die die EALU steuert.

Switching-Tabelle Eine Switching-Tabelle ist ein Ring-Speicher, welcher durch eine Steuerung angesprochen wird. Die Einträge einer Switching-Tabelle können beliebige Konfigurationswörter aufnehmen. Die Steuerung kann Befehle durchführen. Die Switching-Tabelle reagiert auf Triggersignale und konfiguriert konfigurierbare Elemente anhand eines Eintrages in einem Ringspeicher um.

Umkonfigurieren Neues Konfigurieren von einer beliebigen Menge von PAEs während eine beliebige Restmenge von PAEs ihre eigenen Funktionen fortsetzen (vgl. konfigurieren).

Verarbeitungszyklus Ein Verarbeitungszyklus beschreibt die Dauer, welche von einer Einheit benötigt wird, um von einem definierten und/oder gültigen Zustand in den nächsten definierten und/oder gültigen Zustand, zu gelangen.

Zellen Synonym für konfigurierbare Elemente

## 7 Konventionen

### 7.1 Namenskonvention

Baugruppe -UNIT  
Betriebsart -MODE  
Multiplexer -MUX  
Negiertes Signal not-  
Register für PLU sichtbar -PLUREG  
Register intern -REG  
Schieberegisters -sft

## 7.2 Funktionskonvention

### UND-Funktion &

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

### ODER-Funktion #

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

### NICHT-Funktion !

A	Q
0	1
1	0

### TOR-Funktion G

EN	D	Q
0	0	-
0	1	-
1	0	0
1	1	1

## Patentansprüche

1. Verfahren zur Synchronisation und Umkonfiguration von konfigurierbaren Elementen in Bausteinen mit zwei- oder mehrdimensionaler programmierbarer Zellstruktur (DFP, FPGA, DPGA), **dadurch gekennzeichnet**, daß

1. Synchronisationssignal während der Verarbeitung innerhalb des Datenstromes von den verarbeitenden konfigurierbaren Elementen generiert werden und an weitere Elemente zur Synchronisation über den Datenbus gesandt werden,

2. aus dem Datenstrom heraus anhand entsprechender Befehle Konfigurationswörter innerhalb eines programmierbaren Elementes generiert werden und über den Datenbus zusammen mit der Adresse des anzusprechenden Registers an ein weiteres konfigurierbares Element übertragen werden, wobei dieses dadurch ohne den Einfluß einer externen Ladelogik (um)konfiguriert wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß bei der Synchronisation durch einen Trigger ein konfigurierbares Element zur Ausführung einer einzigen Operation angeregt werden kann.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet,

net, daß bei der Synchronisation durch einen Trigger ein konfigurierbares Element zur Ausführung einer Vielzahl Operation angeregt werden kann.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß bei der Synchronisation durch einen Trigger die Ausführung eines konfigurierbaren Elements angehalten werden kann. 5

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß bei der Synchronisation durch einen Trigger ein konfigurierbares Element zur Umkonfiguration freigegeben werden kann. 10

6. Verfahren nach Anspruch 1-5, dadurch gekennzeichnet, daß das konfigurierbare Element seinen momentanen Status in einem Statusregister anzeigt.

7. Verfahren nach Anspruch 1-6, dadurch gekennzeichnet, daß die Angabe der anzusteuenden Register in Befehlen kodiert ist und über den Datenbus übertragen wird. 15

---

Hierzu 2 Seite(n) Zeichnungen

---

20

25

30

35

40

45

50

55

60

65

